MULTISCALE STATISTICAL MODELING APPROACH TO MONITORING MECHANICAL SYSTEMS

Kenneth C. Chou

Applied Control and Signal Processing Group
SRI International
Menlo Park, CA 94025

Abstract

Signal processing for condition based maintenance and equipment monitoring has focused in recent years on non-stationary signal analysis using time-frequency representations of the signal. These representations are used to identify non-stationary events in the signal that indicate some change in the state of a structure or a machine. It is important to be able to reliably detect such changes in real-time to do necessary preventive maintenance and also to minimize unnecessary maintenance. While transformations such as the Wigner-Ville, Gabor, and wavelet transforms are useful in highlighting time-frequency features of the signal, the application of such transforms to the monitoring problem requires additional [illegible] for making decisions concerning the condition of the object being monitored. In particular, the interpretation of the transform coefficients in terms of physical events is essential to making such decisions. We develop a methodology for identifying the physical state of the object based on statistical models of the signals, which could comprise, for example, multiple outputs from devices such as accelerometers, strain sensors, and acoustic emission sensors. Classification of machine states based on monitoring signals is performed by comparing likelihood scores for each machine state. We present examples of applying our system to various data, including damped sinusoids and noisy chirps, as a way of illustrating system performance for the case of transient monitoring signals. We compare our system to one which is trained using a DFT-based (non-time-frequency-based) representation (in particular, LPC coefficients) and show that our system exhibits both superior performance as well as greater robustness to noise in the signals. We also compare results using multiscale parameters versus LPC coefficients for the case of synthesized autoregressive signals and for the case of actual, measured signals from a weld depth monitoring system.

Keywords: multiscale, nonstationary signals, statistical models, time-frequency, wavelets

1 Motivation

In machinery monitoring applications, such as tool wear monitoring for example, the signal representation component of the monitoring system is fundamental to the classification performance of the monitoring system. In these applications, characterization of transient signals is key to classifying machine wear states [9]. Furthermore, the ability to train a pattern recognition system using limited amounts of data is important, especially those encountered in flexible manufacturing. In standard frame-based systems, spectral-based coefficients (e.g. LPC, cepstral, melcepstral parameters) are used to model the data within each frame, the assumption being that this data is stationary. To address the need for characterization of transients and other types of nonstationary signals for machine state classification, work has been done toward using time-frequency representations as a basis for features [10]. In order to address the need for controlling the degrees of freedom of our representation to match the available data (e.g. limited training time, low SNR), we propose the use of a parameterized model for representing the time-frequency characteristics of the data in each frame. By varying the structure and the number of parameters, the model is rich enough to capture a wide range of signals, both stationary and non-stationary. The structure of the parameters, however, can be constrained, e.g. in terms of number of parameters, to accommodate limited training data.

2 Multiscale Models

For nonstationary signals, it has been shown that the statistical correlation between the wavelet coefficients of a process has a great deal of structure [12]. For example, for many processes of interest the strongest

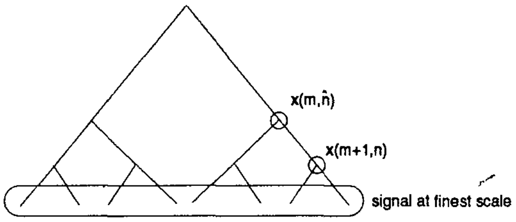

Figure 1. Binary tree representation of a multiscale stochastic process

correlations exist between different scale coefficients located at the same point in time. This type of structure is well-modeled by the class of multiscale stochastic models (MSMs) developed in [1, 2, 3]. These models are represented on a binary tree in which the levels of the tree may be taken to represent projections of the signal onto multiple scales (Figure 1). Our MSM is characterized by the following state equation in which the state, $x(m,n)$, is a vector (this allows for the representation of multichannel data) at the node located at scale m and location n in the binary tree, while $y(m,n)$ is the observed value corresponding to the signal value at that node.

$$\begin{aligned} x(m+1,n) &= F_{m+1,n}x(m,\hat{n}) + w(m+1,n) && (1)\\ y(m+1,n) &= H_{m+1,n}x(m+1,n) + v(m+1,n) && (2) \end{aligned}$$

where $F_{m+1,n}$ is the matrix that maps the state at scale m and location $\hat{n}$ to the state at scale $m+1$ and location n, and $w(m+1,n)$ is the random variable uncorrelated with $x(m,\hat{n})$, representing the detail that is added. The matrix $F_{m+1,n}$ represents, for example, the weighting of one of the branches in Figure 1 in going from a coarse node to one of the finer nodes below it. The matrix $H_{m+1,n}$ and the random variable $v(m+1,n)$ allow for the observed signal value at node m,n to be a linear function of the state plus white noise.

For the purpose of modeling features in a frame of data, the observations at the bottom or finest level of the tree would represent the data in a particular frame. In this case, the observations $y(m,n)$ are non-zero only at the finest level. The state vector $x(m,n)$, however, exists at every node on the tree. By varying $H_{m,n}, C_{m,n}$, the variances of $w(m,n)$, and the variances of $v(m,n)$, we can model a rich class of random phenomena including fractal Brownian motion and Markov random processes [8]. This leads us to investigate the use of the model parameters of our MSM as features for classification of nonstationary signals.

As a final note on multiscale tree models, we summarize the second-order statistical properties of the model and make a comment about parameterization of these models. Note that the following discussion will be in terms of the notation in Figure 2. To begin with, we describe the local covariance $C(t)$ (i.e. the covariance of the state-vector at a particular node t) as it propagates from one node to the next in the following way.

$$C(t) = F(t)C(\gamma t)F^T(t) + P(t)$$

where γ is a backward shift operator on the tree and $P(t)$ is the variance of the white noise term $w(t)$. Note that the covariance of the state at a particular node is a function of the covariance at the parent node dynamically driven by the covariance of the white noise term added at that node.

We now describe the cross-covariance between states at two different nodes, i.e the correlation between the state at two different nodes. For two distinct nodes on the tree, $s \neq t$,

$$\begin{aligned} E[x(s)x(t)^T] &= (\prod_{i\in\Gamma_{us}} F_i)C(u)(\prod_{j\in\Gamma_{ut}} F_j^T)\\ u &= s \wedge t \end{aligned}$$

where $\wedge$ is a function that maps two nodes s and t into their unique common (least) ancestor node, and Γ_{us} is the ordered set of nodes $\{s, \gamma s, \gamma^2 s, \ldots \tilde{u}\}$, where $\gamma\tilde{u} = u$. Referring to Figure 2, note that the cross-covariance between states at nodes s and t is simply a function of the state covariance at the node corresponding to

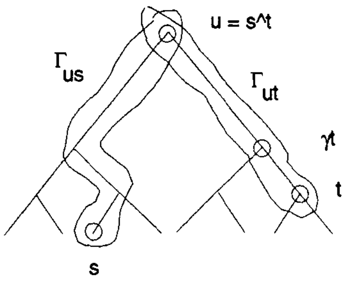

Figure 2. Illustration of dependencies of correlation function on trees

their first common ancestor and the dynamic matrix, F, along the two paths joining s and t, respectively, to their common ancestor node (denoted u in Figure 2).

The correlation characteristics of the MSM's point to several observations. First, if we consider F to be a constant, independent of node location, then the characteristics of the MSM become quite specific. In particular, for example in Figure 2, node s is correlated with each of the four nodes in the right half of the bottom level of the tree in *precisely the same way*. Note that regardless of how the variance at each node, $C(t)$, evolves, if F is constant, the correlation between states at two nodes is simply a function of $C(t)$ at the ancestor node and the number of branches connecting the two nodes to their common ancestor. In [2], it is shown that this correlation structure has eigenfunctions corresponding to the Haar wavelet transform. Clearly, if we want to expand our MSM's to include a wider class of processes, we must consider a richer class of parameterizations.

In this paper, we explore the idea of varying the dynamic matrix F in our MSM's to allow for richer descriptions of signals in terms of their correlation structure. In particular we will show examples of allowing F to vary arbitrarily over nodes starting from the top node down to and including the nodes at some specified scale. Thus, by varying F we can vary the order of our MSM, much in the same way as one would vary the order of a parameterized statistical model such as the class of autoregressive (AR) models. By allowing F to vary across both scale and time we would expect non-stationary behavior, and in particular behavior such as changing spectral characteristics within one's analysis window, to be better captured by our models than they would using stationary models.

3 Multiscale Parameter Identification

We use an efficient Maximum-Likelihood procedure for identifying the parameters of our MSM based on a frame of data. It is based on the Expectation Maximization (EM) algorithm [4] as applied in [5]. The basic idea behind the algorithm is to first perform the expectation step of the algorithm, i.e. computing the expectation

$$E[\mathcal{L}(X, Y, \theta)|Y] \tag{3}$$

using the current MSM parameters θ, where $\mathcal{L}$ is the likelihood of the data in the frame Y and the state variables X given θ. Then, in the next step, $\mathcal{L}$ is maximized with respect to the parameters θ. In iterating this procedure, one is guaranteed to converge to a local maximum of $\mathcal{L}$ and furthermore, at each iteration $\mathcal{L}$ is guaranteed to increase. The expectation step can be performed rather efficiently for our models using efficient smoothing algorithms developed in [2]. These algorithms are based on a Kalman filtering scheme that exploits the parallel and local structure of the scale dynamics of eq.'s(1,2). This smoothing operation can be performed in $O(m^3 logN)$ operations using parallel processing, where N is the data frame length and m is the state dimension. The maximization step is straightforward, resulting in the computation of sums of outer products involving state vectors at each node and observations at the finest scale.

The basic points about the EM algorithm for Maximum-likelihood estimation of MSM parameters can be summarized as follows.

- Expectation step: Compute

$$E[\mathcal{L}(X, Y, \theta)|Y] \tag{4}$$

- Involves computing expectations of MSM state x and local correlations conditioned on data Y
- Fast algorithm exploits Markov structure of model: $O(N)$ complexity, $O(logN)$ with parallel processing

- Maximization step: Compute

$$\overset{max}{\theta}\ E[\mathcal{L}(X,Y,\theta)|Y] \tag{5}$$

  - Direct solution exists
  - Sufficient statistics easily computable, $O(N)$ complexity

3.1 Classifying Features

Once the signal is modeled at the frame level using MSM parameters as features, we need a way of modeling statistically the ensemble of frames for a given signal. We model the sequence of frames of features as samples of a multivariate distribution. Moreover, by doing this we provide a way of mitigating the effects of sensitivity of our multiscale models to time shifts. The Gaussian Mixture Model [11] provides a convenient multimodal model for our ensemble of frames. In particular, the GMM leads to an efficient recognition procedure for classifying signals by processing successive frames of observation vectors and evaluating likelihood functions [6]. Also, we use an efficient procedure for training our GMM based on applying the EM algorithm to the problem of identifying the GMM parameters. This iterative procedure, as was the case for the MSM parameter identification procedure, yields locally optimal, maximum likelihood parameter estimates by ensuring a nondecreasing likelihood function after each iteration.

4 Numerical Examples

We present numerical examples to give an indication of how our system performs as compared to a system using features based on a spectral representation. We will show examples from a variety of data sets, but the basic procedure for analyzing the capabilities of our model will be the following. For a particular type of data (e.g. damped sinusoids), we either synthesize or measure signals from two different classes (e.g. representing two different machine states). We then analyze and compare the classification performance for two different representations: 1) our MSM's and 2) LPC models. To characterize classification performance, we will use bounds on classification error, visualization of classification surfaces, and simulations of classification performance based on actual training using data. Note that unless otherwise stated, our frame size is taken to be 64 samples.

For our first example, we generate data from two different classes of transient signals, representing noisy damped sinusoids. The transients will be modeled as follows for classes $j = 1, 2$:

$$y_j(t) = \sum_{i=1}^{N_j} K_{i_j} e^{-\alpha_{i_j} t} \sin(\beta_{i_j} t) u(t) + \eta_j \tag{6}$$

where N_j controls the density of transients within a given data segment and η is Gaussian white noise. Figure 3 is a plot of typical signals from each of the two classes.

4.1 Bounds on Classification Error

We now examine the performance of a classifier using our MSM parameters as features versus using features based on a spectral representation. For the examples in this section our MSM's are fixed at order 4, i.e. the parameters consist of a fixed dynamic parameter, F, the observation parameter, H, and the two noise variance corresponding to the white noise driving term w, and the white noise observation term v, respectively. In later examples we will allow F to vary in order to demonstrate the capabilities of higher-order MSM's. As an example of spectral features we use Linear Predictive Coding (LPC) coefficients, set at order 4 in order to normalize the number of parameters to be equal to the number of MSM parameters. To give an indication

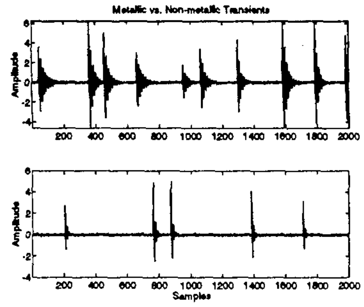

Figure 3. The top plot shows an example of a typical signal in Class 1 and the bottom plot shows a typical signal in Class 2. Note the number of transients in the Class 1 signal is higher, and the damping factor is smaller.

Data scenario	*MSM features*	*LPC features*
noiseless	6.8099e-4	.3227
SNR 12dB	.0072	.4161
SNR 6dB	.0083	.4519

Table 1. Comparison of Bhattacharyya bounds using MSM versus LPC features for transient data. 4000 sample sequences used.

of the performance of a classifier based on a particular set of features, we using Bhattacharyya bounds [7] to estimate the classification error. The Bhattacharyya bound gives an upper bound on the probability of error in a 2-class classification problem. We can compute the bound using sample means and sample covariances of the set of features derived from the two classes of signals. Table 1 contains bounds on the classification error probability for the 2-class data set in Figure 3 for the case of MSM features versus LPC features. Results are shown for the noiseless case as well as for SNR's of 12dB and 6dB. From these results we see not only that MSM features leads to better discrimination between the two types of transients but also that they seem to be less sensitive to noise than the LPC features.

We now show performance comparisons based on other data sets. Table 2 shows results for two different data sets. One data set consists of signals from a weld penetration monitoring system. Figure 4 shows optical signals for the case of a full penetration weld and a partial penetration weld. Note that the signals differ quite subtly in character and that the MSM parameters seem to discriminate better than the LPC parameters do. Also in Table 2 are results on synthesized autoregressive data. Two classes of AR(1) processes were created with lag coefficients of .97 and .9, respectively. Perhaps surprisingly, the MSM features outperform the LPC features, even though the signals are synthesized to be stationary and are of the LPC model class. However, it is important to note that we are measuring discrimination performance rather than representational

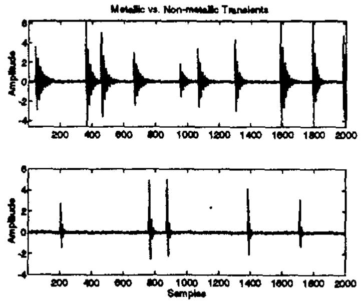

Figure 4. The top plot shows an optical signal corresponding to a full penetration weld while the bottom plot shows an optical signal corresponding to a partial penetration weld.

Data scenario	*MSM features*	*LPC features*
weld depth data	.2890	.4371
AR process	.2230	.3835

Table 2. Comparison of Bhattacharyya bounds using MSM versus LPC features for optical weld depth signals, AR(1) processes. 4000 sample sequences used.

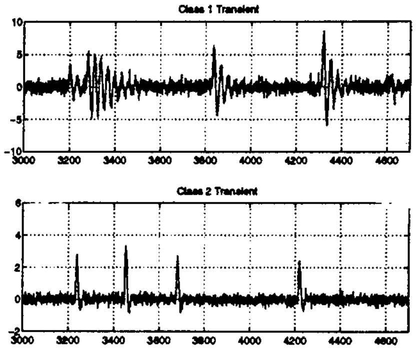

Figure 5. Transient signals

performance.

4.2 Visualization of Classification Surfaces

In this section we provide further examples of differences in classification performance between using MSM's and using LPC models, and do a comparison of performance for different order MSM's and LPC models. In addition to bounds on classification error, we provide in the following a visualization of the actual estimated densities of the MSM and LPC parameters for each of the two classes corresponding to a particular data set. In particular, we compute the sample means and covariances of the features (both MSM and LPC) for each class and present a 3D plot of the modes corresponding to these means and covariances for these two classes. Since we can only visualize modes of at most 2 features, we project the multivariate modes onto 2 features representing: a) the 2 features giving the best mode separation, corresponding to best classification and b) the 2 features giving the worst mode separation, corresponding to worst classification.

Figures 5-7 show sample paths for two classes of signals corresponding to three different data types: 1) transient signals generated as before using noisy damped sinusoids 2) 6th order AR signals and 3) chirp signals. The plots of the modes based on sample means and covariances are depicted in Figures 8-10. Note that in each case the mode separation is greater, indicating better classification performance, in the case of MSM parameters versus the case of LPC parameters. This is consistent with the bounds computed for these examples as given in Table 3.

Table 3 also gives bounds for the case of 6th order and 10th order MSM and LPC models applied to the AR signals as well as the chirp signals. It is interesting to note that even in the case of the AR signals, one in which we might expect the LPC model to do better, the MSM features do a much better job of classification based on the bounds. This may be due to a much greater sensitivity on the part of LPC models to edge effects due to small window sizes. Note that in the case of the chirp signals we compare MSM and LPC performance for orders 4-6 for the case of a larger window (256 samples). For this particular window size, LPC does better than MSM for 4th order but worse for orders 6 and 10. Of course, the bounds can be deceiving and thus, we plot the modes for the order 6 case in Figure 11 (best two features). The 6th order MSM features show considerably more mode separation than the 6th order LPC features.

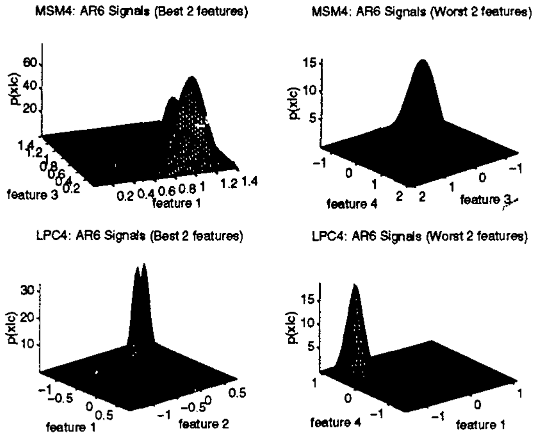

Figure 9. Single mode densities for AR6 signals

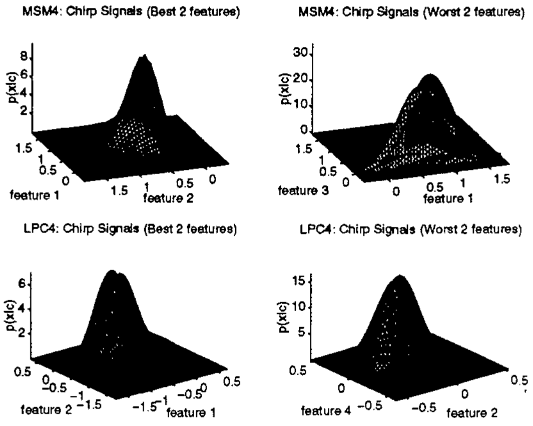

Figure 10. Single mode densities for chirp signals

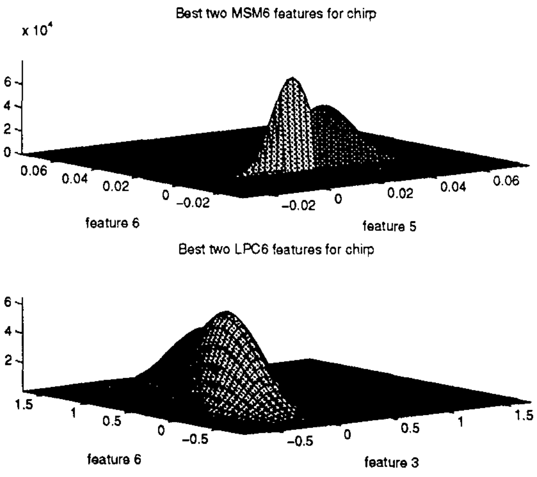

Figure 11. Single mode densities for chirp signals: 6th order models

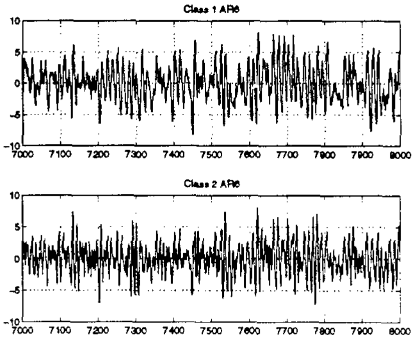

Figure 6. Autoregressive signals

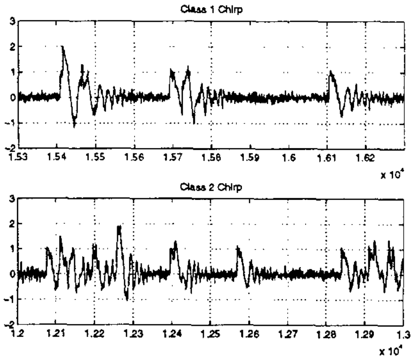

Figure 7. Chirp signals

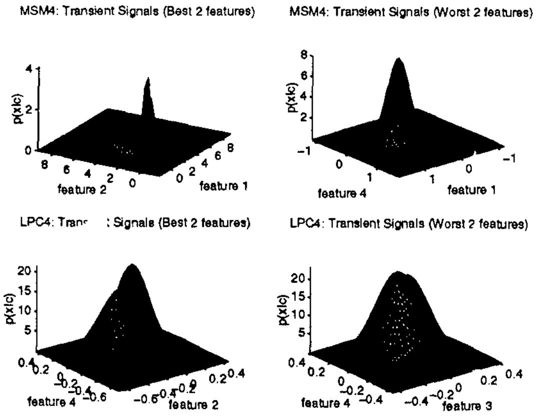

Figure 8. Single mode densities for transient signals

SIGNAL TYPE	MSM 4	LPC 4	MSM 6	LPC 6	MSM 10	LPC 10
Transients	.1387	.4709				
AR 6 Processes	.1670	.1231	7.1639e-10	.1552	1.7498e-19	.2047
Chirps	.3636	.4723	.0057	.4620	.3711	.4602
Chirps (256 pt. frame)	.4040	.3628	.3346	.3570	.0515	.3345

Table 3. Comparison of Bhattacharyya bounds using MSM versus LPC features for three signal types, orders 4,6 and 10.

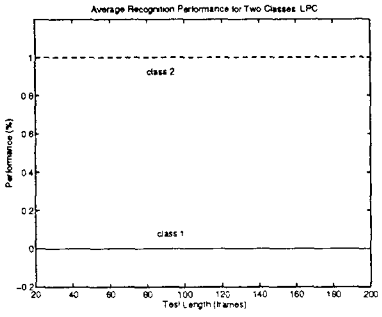

Figure 12. LPC performance curve

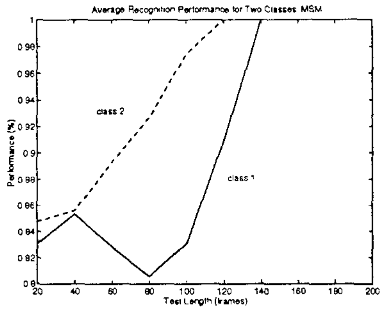

Figure 13. MSM performance curve

4.3 Classification performance based on simulation of classifiers

Finally, we show the results of training a classifier based on using MSM parameters as features versus using LPC parameters as features. We train both a 4th order MSM model and a 4th order LPC model on 24,000 samples of each class of chirp signals. To do the training, a 64 sample frame is used in each case, each frame giving rise to 4 features corresponding to the MSM (LPC) parameters for that frame. Furthermore, we use a 3-mode Gaussian mixture model to model the features over the entire collection of frames. We then use an independent set of 24,000 samples of each class of chirp signal to perform a classification test using MSM features versus using LPC features. The classification is performed by doing a likelihood ratio test based cn the Gaussian mixture model. Figures 12,13 give performance plots for tests given each chirp signal type, using both MSM and LPC features. Average performance is given for different amounts of data used to perform the likelihood ratio, where each point on the curve reflects average detection performance using a fixed number of frames. For example, for test length equal to 100, the performance value is equal to the percentage of correct classifications for a particular signal class, computed over the entire 24,000 sample data set, using 100 frame chunks to make each classification decision. In the case of LPC features, for this particular set of data the classifier chooses class 2 every time, yielding .5 probability of error, independent of the number of frames used to make the decision. This of course indicates the worst possible performance (note that the bound indicated .4723 as an upper bound on the probability of error; violation of this bound is due to finite-sample effects). For the MSM features, performance starts at around .82 probability of detection and eventually converges on zero percent error. Again finite-sample effects come into play, especially when the ratio of the number of frames used to make the decision to the total number of test samples becomes large.

5 Acknowledgment

We would like to thank Ken Pietrzak at UTRC for providing the weld penetration monitoring data.

References

[1] K.C. Chou, S.A. Golden, and A.S. Willsky, "Multiresolution Stochastic Models, Data Fusion, and Wavelet Transforms," *Signal Processing*, Vol. 34, Issue 3, 1993.

[2] K.C. Chou, A.S. Willsky, and A. Benveniste, "Multiscale Recursive Estimation, Data Fusion, and Regularization," *IEEE Trans. on Automatic Control*, Vol. 39, No. 3, March 1994.

[3] K.C. Chou, A.S. Willsky, and R. Nikoukhah, "Multiscale Systems, Kalman Filters, and Ricatti Equations," *IEEE Trans. on Automatic Control*, Vol. 39, No. 3, March 1994.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM algorithm," J. R. Stat. Soc. Lond., Vol. 39, pp. 1-38, 1977.

[5] V. Digalakis and K.C. Chou, "Maximum Likelihood Identification of Multiscale Stochastic Models Using the Wavelet Transform," *Proc. of ICASSP*, Minneapolis, MN, 1993.

[6] L.P. Heck and K.C. Chou, "Gaussian Mixture Model Classifiers for Machine Monitoring," *Proc. of ICASSP*, Adelaide, Australia, 1994.

[7] T. Kailath, "The Divergence and Bhattacharrya Distance Measures in Signal Selection," *IEEE Trans. of on Communication Tech.*, vol. Com-15, no. 1, Feb. 1967.

[8] M. Luettgen, W. Karl, A. Willsky, and R. Tenney, "Multiscale Representations of Markov Random Fields," *IEEE Trans. Signal Processing*, Vol. 41, No. 12, December 1993.

[9] S. Rangwala and D. Dornfeld, "Sensor Integration Using Neural Networks for Intelligent Tool Condition Monitoring," *Trans. of ASME, Journ. of Eng. for Industry*, Vol. 112, No. 3, pp. 219-228

[10] R. Priebe and G. Wilson, "Application of 'Matched' Wavelets to Identification of Metallic Transients," *IEEE-SP Symp. on Time-Freq. and Time-Scale Anal.*, Victoria, B.C., October 1992.

[11] D.A. Reynolds, *A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification*, Ph.D. Thesis, Georgia Institute of Technology, September 1992.

[12] A. Tewfik and M. Kim, "Correlation structure of the wavelet coefficients of fractional brownian motions," *IEEE Trans. Informat. Theory*, vol. 38, Mar. 1992.